Prototypical Networks for Few shot Learning

Ishrat Jahan Ananya

November 17, 2022

Abstract

The report details a brief analysis of the Prototypical Networks algorithm for Few Shot Learning in Image classification tasks. The referenced paper was studied in depth and multiple experiments are presented to highlight the algorithm's key features.

1 Problem Description

Few Shot Learning is the next generation class of algorithms being researched currently for object categorization tasks in Computer vision with unseen data and without pre-training.There are many such architectures available such as 'Matching networks for one shot learning'[1], 'Learning to compare: Relation network for few-shot learning' [2] etc. For this project, the task is to implement the Prototypical Networks algorithm for few shot learning [3] problems.

2 Methodology

2.1 Foundational Concept

In few shot classification, it is essential for the model to accommodate the new unseen classes with limited examples to train on. The key feature of the paper 'Prototypical Network for Few Shot Learning' is that they introduced the concept of prototypes and calculating distances of example images from these prototypes in order to classify the newly introduced images. Prototypes are simply

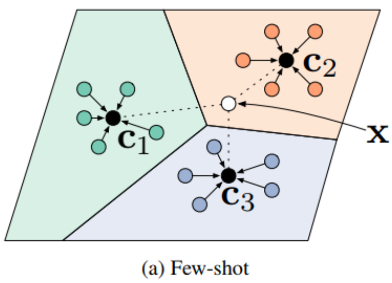

Figure 1: Few Shot Learning

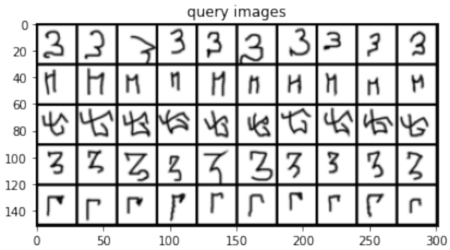

Figure 2: Omniglot data

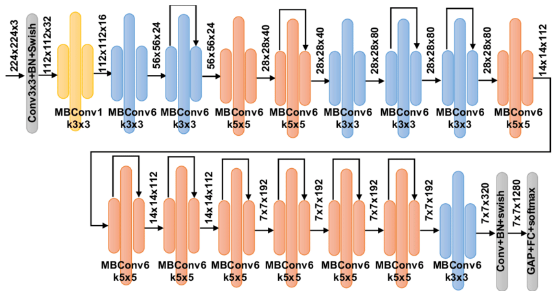

Figure 3: Efficient Net architecture

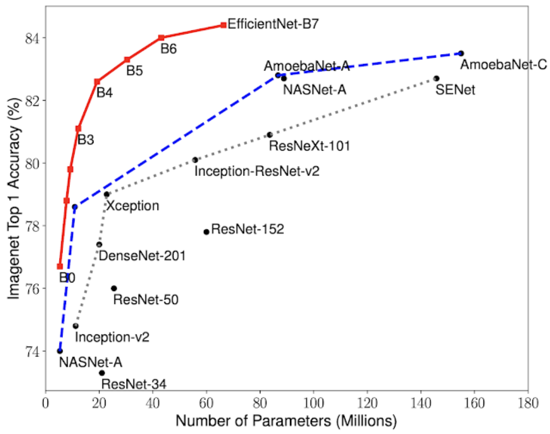

Figure 4: EfficientNet performance in comparison to other models

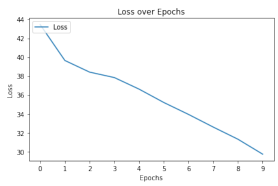

Figure 5: Experiment 1 Loss vs epoch

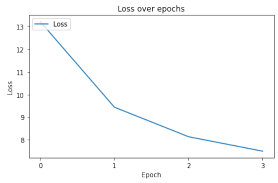

Figure 6: Experiment 2 Loss vs epoch

a mean embedding representation for all samples in each class that is mapped by using a Convolution Neural Network Architecture as a feature extractor. Classification can then be implemented by calculating the euclidean distance between the unknown image and the nearest class prototype as demonstrated in Fig 1. The main intuition here is that the calculated mean of all samples in a class is the best possible single representation that defines the class in question very well. This idea is leveraged in the paper and it has yielded in State of the Art results.

To discuss the problem at hand even further,there are two sets of input images: the Support set with its labels and the query set whose labels are to be predicted. The model is trained on the few samples present in the support set and then use the query set for prediction and evaluation purpose. Another new concept introduced called episodal training affects the training process of the few shot learning problem. The idea is to train the model on a specified number of tasks called episodes. Each epoch can have thousands of episodes. This special arrangement of data and episodes can further be controlled by indicating how many classes and samples are present in the support set of the dataset. If there are 5 classes, each with 3 example images in the support set , then we can refer to it as a 5-way 3 shot classification problem. It is highlighted in the paper that an ideal scenario includes matching the number of shots to the number of ways specified for a given problem to get better performance.

2.2 Dataset

The paper conducted a few experiments with multiple datasets such as a mini version of Image-net called the Miniimagenet dataset [4], Omniglot dataset [5] and CUB dataset [6]. For our project we used the Omniglot dataset only which is available in TorchVision. There are 1623 handwritten characters collected from 50 alphabets. There are 20 examples associated with each character, where each example is drawn by a different human subject.The dataset has black and white images meaning it has only 1 channel, so during preprocessing,they were changed to have 3 channels and all the images were resized and horizontally flipped. Few samples images from the dataset are shown in Fig 2.

To load the dataset in a custom dataloader, EasyFSL [7] library was used that arranged the dataset into torch dataloader in such a way that, it became easy to retrieve the support images, support labels, query images and keep track of the episodes to be trained on, given the image path, number of shot, number of query images to retrieve and number of way is pre-specified. The library extends the existing Task Sampler class available in Pytorch to provide the above mentioned functionality.

2.3 Implementation

The paper offers an upgrade over previous methods discussed by utilising euclidean distance to estimate the similarities between prototypes and query images. It claims that the euclidean method performs better than using cosine similarity. Furthermore, the paper uses a simple convNet as a feature extractor to map the support images in the embedding space.The ConvNet is comprised of a few convolutional blocks paired with pooling and batch normalization layers.

For our project, we have experimented with a different backbone and a different distance measure in order to demonstrate better performance as opposed to the original paper. We have used pretrained EfficientNet [8] model in the backbone of our Network. The justification lies in the fact that Efficientnet is a relatively larger architecture that is capable of richer feature extraction that allows for a more reliable embedding space. Moreover, it is much more computationally efficient in comparison to other networks such as Res-net, Resnext or VGG. Fig 3 and 4 show the architecture of efficient net and the performance comparison with other models respectively.

After loading the model pretrained on Imagenet, the feature extractor is passed to a ProtoNet Class which essentially calculates the prototypes for all classes present in the support images. Next we calculate the Manhattan distance between the prototypes and query images instead of using the euclidean distance. The Manhattan distance is defined as the sum of the absolute differences of their Cartesian coordinates. This distance is usually a better choice when there is high dimensionality in the data[9]. The formula for the distance calculation is given as follows:

$$Dist_M = \sum_{i}^{n} |X_i - Y_i|$$

First a 3 way 3 shot network was trained with 100 training episodes and 10 epochs. For loss , Cross entropy loss was used which is the most appropriate option as it is a multi class classification and for optimiser, Adam optimiser was picked with a learning rate of 0.001. The choice of using Adam over Stochastic Gradient Decent as used in the original paper is because Adam is better at convergence and efficiency during training.

Second experiment took this same concept a step further and trained a 5 way 5 shot classification model over 1500 episodes and only a few epochs. We kept the loss and optimiser constant for valid comparison.

3 Results

The first experiment took mean of loss for all episodes in each epoch and yielded a performance accuracy of 73 percent. Figure 5 shows the loss drop of the model over epochs.

The Second experiment implemented a 5 way 5 shot classification task with 1500 episodes per epoch and gave a performance accuracy of 88 percent.Figure 6 shows the loss drop of the model over epochs.

The experiments demonstrates that if the model could be trained for much longer such as more than hundred epochs and more than two thousands episodes as done in the original paper the accuracy could be pushed further up. Yet, with the limited training it received along with the extremely few samples used for training, the model showed promising results. If a traditional training approach was acquired it would have severely overfit on this problem due to the few number of support images. This proves the usefulness of few shot learning and its massive potential to future classification tasks on limited data such as One shot or Zero shot Learning where the support set is further constrained.

4 References

1. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D. (2016). Matching networks for one shot learning. Advances in neural information processing systems, 29.

2. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1199-1208).

3. Snell, J., Swersky, K., Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in neural information processing systems, 30.

4. https://paperswithcode.com/dataset/mini-imagenet

5. https://paperswithcode.com/dataset/omniglot-1

6. https://paperswithcode.com/dataset/cub-200-2011

7. https://github.com/sicara/easy-few-shot-learning

8. Tan, M., Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

9. https://medium.com/@kunal$_{g}ohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7$